Videojuegos

Curso de Diseño y Programación

Nº 4 5,99 euros

*Trabajar con el editor de sonido **Goldwave***

*Sentencias principales de **Blitz 3D***

*Manejo del secuenciador de sonido **Cubase VST***

MULTIPLAYER SETUP

PLAYER SETUP







Videojuegos

Curso práctico de Videojuegos

AUTOR DE LA OBRA
Marcos Medina
DIRECCIÓN EDITORIAL
Eduardo Toribio
etoribio@iberprensa.com
COORDINACIÓN EDITORIAL
Eva-Margarita García
eva@iberprensa.com
DISEÑO Y MAQUETACIÓN
Antonio Gª Tomé
PRODUCCIÓN
Marisa Cogorro
SUSCRIPCIONES
Tel: 91 628 02 03
Fax: 91 628 09 35
suscripciones@iberprensa.com
FILMACIÓN: Fotpreim Duvial
IMPRESIÓN: Gráficas Don Bosco
DUPLICACIÓN CD.ROM: M.P.O.
DISTRIBUCIÓN
S.G.E.L.
Avda. Valdelaparra 29 (Pol. Ind.)
28108 Alcobendas (Madrid)
Tel.: 91 657 69 00

EDITA: Iberprensa
www.iberprensa.com

CONSEJERO
Carlos Peropadre

REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN
C/ del Río Ter, 7 (Pol. Ind. "El Nogal")
28110 Algete (Madrid)
Tel.: 91 628 02 03
Fax: 91 628 09 35
(Añada 34 si llama desde fuera de España.)

DEPÓSITO LEGAL: M-35934-2002
ISBN: Coleccionable: 84 932417 2 5
Tomo 1: 84 932417 3 3
Obra Completa: 84 932417 5 X

NOTA IMPORTANTE:
Algunos programas incluidos en los CD de "Programación y Diseño de Videojuegos" son versiones completas, pero en otros casos se trata de versiones demo o trial, versiones de evaluación que Iberprensa quiere ofrecer a nuestros lectores. No se trata en ningún caso de las versiones comerciales de los programas, y las hemos incluido para dar al lector la oportunidad de conocer y probar esos programas y que así pueda decidir posteriormente si desea o no adquirir las versiones comerciales de cada uno.

Aprende divirtiéndote

Bienvenidos a **Programación y Diseño de Videojuegos**, la primera obra coleccionable cuyo objetivo es formar al alumno en las principales técnicas relacionadas en el desarrollo completo de un videojuego.

A lo largo de la obra el lector aprenderá programación a nivel general y a nivel específico con ciertas herramientas y lenguajes, aprenderá a trabajar con aplicaciones de retoque de imagen y también de diseño 3D y animación. Descubrirá las aplicaciones profesionales más importantes de audio y conocerá la historia de lo que se denomina "la industria del videojuego", los últimos 20 años, los juegos que marcaron un avance, sus creadores y en general la evolución del videojuego.

Pero además, esta obra tiene un segundo objetivo, desarrollar y potenciar la creatividad del lector, nosotros a lo largo de las diferentes entregas pondremos las bases y tú pondrás tu ingenio, tu creatividad y tu capacidad de mejorar.

Comienza aquí un viaje de 20 semanas articulado en 400 páginas y 20 CD-ROMs cuya finalidad es proporcionar las bases mínimas para después cada uno continuar su camino.

Recuerda que para alcanzar el éxito necesistas cumplir tres condiciones: que te gusten los juegos, poseer cierta dosis de creatividad y finalmente capacidad de estudio.

Una la cumples seguro.

sumario



PARA ENCUADERNAR LA OBRA:

- *Para encuadernar los dos volúmenes que componen la obra "Programación y Diseño de Videojuegos" se pondrán a la venta las tapas 1 y 2.*
- ***Tapas del volumen 1 ya a la venta.***
- ***Los suscriptores recibirán las tapas en su domicilio sin cargo alguno como obsequio de Iberprensa.***

SERVICIO TÉCNICO:
Para consultas, dudas técnicas y reclamaciones Iberprensa ofrece la siguiente dirección de correo electrónico: games@iberprensa.com

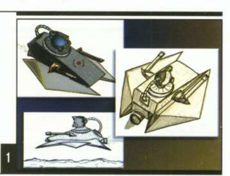
PETICIÓN DE NÚMEROS ATRASADOS:
El envío de números sueltos o atrasados se realizará contra reembolso del precio de venta al público más el coste de los gastos de envío. Pueden ser solicitados en el teléfono de atención al cliente 91 628 02 03

Diseño de los elementos II: los **personajes**

En el número anterior desarrollamos la idea global de nuestro juego. Realizamos guiones y *storyboards*, dibujamos bocetos y describimos el contenido de la acción. Ya ha llegado el momento de definir cada componente de *Zone of Fighters* con más detalle. Vamos a poner definitivamente en práctica toda la teoría, empezando por nuestro personaje.

(■) EL PROTAGONISTA PRINCIPAL

El protagonista principal diremos que representa el elemento que controla el jugador directamente mediante datos de entrada tomados del teclado, ratón o cualquier otro dispositivo de juego. En nuestro caso, lo que controlamos y vemos en pantalla es un vehículo futurista, una bio-nave de combate; sin embargo, nuestro personaje no es ese vehículo, sino quien lo conduce. De hecho, la historia del juego cuenta que se lucha en naves de combate y no cuerpo a cuerpo. La particularidad del diseño del vehículo hace que se denomine "bio-nave", ya que el guerrero está fusionado físicamente con los controles de la nave. Y para que este concepto tenga un significado visible en el juego, se ha mantenido la cabeza del guerrero (el protagonista principal) en el exterior, recubierta por un casco que está conectado por un tubo "biológico" al casco de la nave (ver fig. 1).



1

El protagonista está conectado biológicamente a la nave.

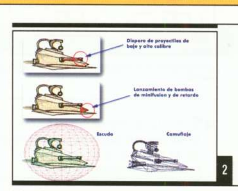
El porqué de este diseño tiene su explicación en la riqueza de visualización que puede ofrecer.

El formato del Blitz3D .B3D, el cual utilizaremos para nuestra "bio-nave", es muy apto para animar modelos orgánicos por la sensación de realidad que ofrece, pero no es nuestro caso. Un vehículo no tiene muchas animaciones con movimientos que puedan dar alguna plasticidad. Sacar la cabeza fuera de la nave e incluir el "bio-tubo" nos permitirá animar el giro de ambos, dando así la sensación de que alguien pilota la nave. Podríamos incluir algunas animaciones más en el modelo como, por ejemplo, el retroceso de los cañones al disparar o mover las alas en los cambios de dirección.

(■) TIPO DE ACCIONES

Básicamente, las acciones que puede realizar una bio-nave están relacionadas con el combate, la defensa y el desplazamiento.

En cuanto al combate y la defensa, puede realizar disparos con sus dos cañones, activar un escudo de protección y camuflarse con el entorno (ver fig. 2).

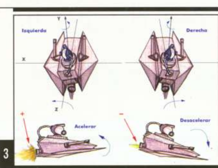


Tipos de acciones posibles que puede realizar la bio-nave.

En cuestión de movimiento, la bio-nave se desplaza en suspensión, flotando a dos o tres metros de la superficie. Por ello, las acciones de aumentar y disminuir velocidad, así como las de cambio de sentido serán suaves y con inercia. Asimismo, para dar un poco más de realismo a la física, aplicaremos por cada cambio de movimiento una ligera inclinación a la nave. A todo esto le sumamos (ver fig. 3):

- **1.** Giro de la cabeza a la izquierda o a la derecha de forma aleatoria mientras la nave se desplaza.

> **NOTA**
>
> **El formato .MD2 funciona por "Deformación de Malla", consiguiendo una animación suave de las partes de un modelo. Además puede incluir multitud de animaciones distintas en el mismo fichero, sin embargo un modelo en este formato no puede ser detectado por los comandos de colisones del Blitz3D. Por este motivo utilizaremos el formato estándar. B3D, el cual emula perfectamente el funcionamiento de los .MD2 y sí puede ser detectado.**



La inclinación de la nave aporta un poco más de realismo.

■ **2.** En la aceleración aumenta la combustión que sale de la tobera, y al tomar impulso la nave se inclina ligeramente hacia atrás por la inercia.

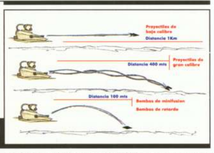
■ **3.** En la desaceleración ocurre lo contrario con la combustión, y la nave, con el impulso, se inclina hacia delante.

A continuación, se muestra una relación más definida de las acciones en el combate de la bio-nave:

■ **1.** Está equipada con dos cañones de munición intercambiable; esto quiere decir que puede disparar proyectiles de diferente calibre. Los distintos tipos de munición están repartidos por todo el terreno de combate.

■ **2.** Además ofrece la posibilidad de lanzar bombas de retardo, las cuales explotan al cabo de unos segundos después de caer al suelo.

■ **3.** Posibilidad de lanzar bombas de minifusión.



Comportamiento de los distintos tipos de munición.

■ **4.** Está dotada de un sistema de camuflaje que la hace invisible durante cierto tiempo.

■ **5.** Capacidad de activar un escudo de protección para repeler ataques durante un tiempo definido.

Los distintos tipos de munición están repartidos por todo el terreno de combate, así como las bombas de retardo. La energía necesaria para activar los sistemas de camuflaje y defensa es limitada, aunque puede ser recargada recogiendo módulos de energía repartidos por el terreno.

(■) TIPO DE ARMAS

Seguidamente, vamos a describir qué clases de armas puede disparar la bio-nave de combate. Realmente, y como ya apuntamos, dispone de dos cañones de munición intercambiable y de un lanzador de bombas de retardo. La cuestión es describir el comportamiento de los distintos tipos de munición (ver figura 4):

■ **1.** Proyectiles de bajo calibre. Es la munición estándar. Tienen gran alcance y permiten una frecuencia de disparo elevada, pero su poder destructivo es reducido. No necesita ser recargada, por lo tanto su número es infinito.

■ **2.** Proyectiles de gran calibre. Son como pequeños misiles con mayor poder destructivo pero con menor alcance que la munición estándar. Cada recarga tiene un número limitado de 20 unidades, es decir, 10 disparos, ya que cada cañón dispara uno.

■ **3.** Bombas de minifusión. Son pequeñas bombas atómicas de 1/10 de megatón cada una. Al ser lanzadas adquieren trayectoria curvilínea ascendente-descendente. Gran poder destructivo pero con poco alcance. Cada módulo de recarga admite sólo dos unidades cada vez.

■ **4.** Bombas de retardo. Son granadas con activación temporal, explotan a los pocos segundos de ser soltadas en el aire. Cada módulo puede contener hasta 5 unidades.

Hasta aquí hemos descrito todas las cualidades de la bio-nave de combate que conducirá nuestro protagonista. El siguiente paso será describir todos los enemigos que podemos encontrarnos en el terreno de combate, aparte de los demás guerreros.

Puede ocurrir que otros contrincantes no estén guiados por jugadores humanos, sino conducidos por el ordenador. Para este caso y debido a su extensión, dedicaremos un especial estudio en una posterior entrega.

(■) LOS MALOS DE LA PELÍCULA

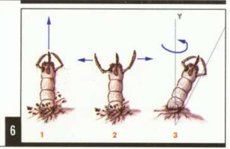
Para dar un poco más de emoción y realismo a la zona de combate, decidimos añadir animales y plantas con comportamientos letales. Procederemos a describir estos seres y su comportamiento.

(■) ANIMALES

Básicamente, sólo incluimos un tipo de animal, lo llamaremos *Slunk*. Son gusanos que habitan bajo la superficie. Son muy peligrosos y aunque son ciegos, tienen un gran olfato y la capacidad de detectar a las bio-naves por la vibración que producen los motores y



Los Slunk son gusanos muy peligrosos que habitan bajo tierra.

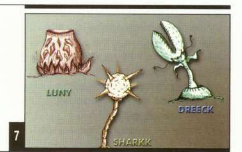


Modo de actuar del terrible Slunk.

las explosiones originadas por el combate. Cuando detectan a una presa, salen a la superficie con gran velocidad abriendo sus grandes tenazas. A veces, también salen engañadas por la acción de las bombas, aunque a menudo se dejan ver olisqueando el aire para buscar comida. Han aprendido que con frecuencia la superficie se llena de otro tipo de habitantes (ver fig. 5).

Determinamos pues una inteligencia inferior con un comportamiento bien definido. En momentos aleatorios el *Slunk* saldrá a la superficie, siempre dentro de su hábitat, en tres ocasiones: En una primera, sin motivo aparente y con alguna frecuencia en lugares imprevistos. En una segunda, aparecerá por casualidad junto a alguna que otra explosión. Y en una tercera ocasión, en la trayectoria de la bio-nave.

En todas estas ocurrencias, el *Slunk* hará la misma acción: abrirá sus tenazas y hará un giro en círculo mientras las cierra, para luego introducirse de nuevo en la tierra (ver fig. 6).



Tipos de plantas peligrosas.

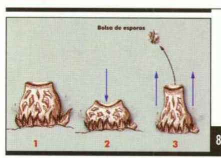
Posee un gran poder destructivo, y un choque en varias ocasiones con el ser puede provocar la total destrucción de la bio-nave. Tampoco es fácil liquidarlo en un enfrentamiento, dado su gran tamaño y la composición blanda de su cuerpo. Aun así, sucumbe bajo el impacto de algunos proyectiles de bajo calibre y con facilidad con un par de gran calibré o cualquiera de las bombas.

Su hábitat preferido es el subsuelo arenoso, debido a la capacidad de movimiento que permite este tipo de material de composición blanda. Sin embargo, y en contadas ocasiones, puede moverse bajo la superficie arenosa del fondo acuático, ya que su respiración cutánea le permite filtrar el oxígeno del agua, así que puede aparecer por cualquier superficie acuática que se encuentre dentro de su territorio.

(■) PLANTAS

En este tipo de seres, incluimos algunas especies que, por su naturaleza, pueden provocar serios daños e incluso la destrucción de nuestra bio-nave. Empezando por los árboles que, dada su robustez, un choque contra ellos dañaría nuestro fuselaje. Sin embargo, hay tres clases de plantas que son particularmente peligrosas: los *Lunys,* los *Sharkk* y las plantas carnívoras *Dreecks* (ver fig. 7).

Los Lunys son plantas que habitan en cualquier superficie arenosa o rocosa. No son carnívoras, pero las hemos incluido como letales por las características especiales de su forma de reproducción. Con frecuencia se contraen para lanzar al aire bolsas gigantes cargadas de esporas a una gran distancia (ver fig. 8). El peligro de esta acción radica en que estas bolsas van car-

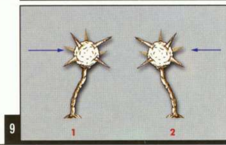


Lanzamiento de bolsas de esporas de un Lunys.

gadas con energía eléctrica con el fin de mantener las esporas unidas. Cualquier intento de separarlas provocará una reacción eléctrica en cadena. Así que un choque con estas bolsas provocará daños en la estructura y en los sistemas de navegación pudiendo incluso hacernos perder el control momentáneo de la nave al cortarse la comunicación biológica entre ésta y el guerrero. Con mucha frecuencia alargan su cuerpo para obtener alimento del aire. Es entonces cuando resultan casi invulnerables, debido a la flexibilidad y resistencia de su cuerpo. Un choque con su piel, también cargada de electricidad, provocará la destrucción completa de la bio-nave.

Estos seres tienen un comportamiento bastante simple. De forma aleatoria y sin ningún motivo, se contraen para luego estirarse y lanzar la bolsa de esporas. Estas bolsas ascienden a una altura nunca igual y caen a una distancia no superior a 200 metros alrededor de la planta.

Nuestro segundo tipo de vegetal animado son los *Sharkk* o plantas martillo. Son de considerable altura y muy abundantes, tanto que a veces pueden formar pequeños bosques. Su estructura es muy particular. Debido a que su delicado sistema reproductivo está ubicado en el centro de la estructura esférica que está sobre el tallo, posee algunos métodos defensivos para evitar cual-

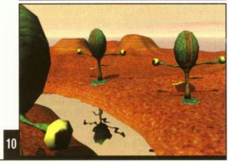


Los feroces Sharkk o plantas martillo pueden destruirnos por completo.

quier contacto ajeno con esta estructura. En primer lugar, unos terribles pinchos la rodean provistos de un sistema de detección de vibraciones en el aire. Al menor movimiento del aire a su alrededor, provocará una expansión y contracción del tallo, que origina un golpe en forma de martillo. Un impacto con los pinchos de esta planta provoca daños en la estructura, y varios golpes seguidos pueden causar la destrucción completa de la bio-nave (ver fig. 9).

Son fáciles de destruir, basta con algunos disparos de bajo calibre, pero la expansión de las esporas recluidas en su interior puede provocar un crecimiento masivo en un lugar cercano.

Nuestra última criatura es quizás la planta más peligrosa: los *Dreecks*. Estas plantas son carnívoras, se alimentan de todo lo que pase por su lado. En realidad son un híbrido entre animal y planta. Poseen tallo y hojas, como todo vegetal, pero están provistas de ojos, boca, estómago, y lo peor de todo: cerebro. A pesar de su gran tamaño, tienen una agilidad increíble y siempre están en movimiento. Al ser también plantas, dependen de las raíces para sustentar su parte vegetal, por eso no pueden desplazarse. Sin embargo, su parte animal tiene un apetito terrible y están en continua búsqueda de presas, girando el tronco y moviendo los apéndices oculares (ver fig. 10).



Los Dreecks son plantas carnívoras muy peligrosas por su voraz apetito.

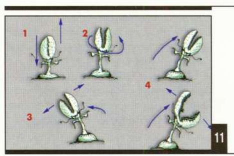
Su posible inteligencia se limita a la posibilidad de describir los elementos que les rodean. Así, pueden seguir la trayectoria de un objeto en movimiento o acechar a cualquier cosa nueva que no esté grabada en su cerebro, la cual consideran como posible alimento.

Sabiendo esto, podemos definir sus movimientos de la siguiente forma:

- **1. Reposo.** Movimientos de arriba a abajo con leves inclinaciones de cabeza.
- **2. Búsqueda de presas.** Mueve el tronco de un lado a otro a la misma vez que mueve sus tentáculos oculares.
- **3. Acecho.** Inclinado y abriendo y cerrando la boca.
- **4. Ataque.** Movimiento brusco del tronco hacia la presa a la vez que realiza un gran mordisco (ver fig. 11).

Un tropiezo con esta criatura letal significa la muerte en el acto. La bio-nave quedará hecha añicos irremediablemente. Destruirla requiere un gran esfuerzo de munición, así que lo más sensato en caso de toparse con ella es esquivarla.

Si bien suelen vivir más o menos en grupos, los individuos están algo distanciados entre sí, y algunas veces se encuentran totalmente solos en lugares dispares fuera de su hábitat normal. Prefieren la arena, aunque no les desagradan las zonas verdes o las laderas de los ríos y lagos.



Movimientos de ataque de los Dreecks.

Ya tenemos, de una forma bien definida, el aspecto y todo el comportamiento de nuestra bio-nave y de los demás habitantes de la zona de combate. Sabemos dónde viven los animales y plantas, qué terreno prefieren, qué daños causan y qué resistencia tienen y cómo se comportan.

Todo este diseño nos servirá para poder darles vida luego en programación, además de saber dónde situarlos en el momento de construir el escenario del juego.

(■) OTROS ENEMIGOS

Para añadir aún más emoción y dinamismo al juego, se podría añadir otro tipo de obstáculos enemigos como artefactos voladores que participen en los combates. Estarían esparcidos por todo el terreno y siempre al acecho. Los llamaremos "voladores". Poseen armas de bajo calibre y alcanzar a uno de estos aparatos proporciona puntos extras.

En el próximo número...

... seguiremos diseñando los elementos que componen *Zone of Fighters*, y le llega el turno a todo lo concerniente al entorno del juego: ambientación, decorados, objetos de adorno y comportamiento de los objetos especiales que nos ayudarán a combatir.

Modelado en baja poligonización con **Milkshape3D** (II)

En el capítulo anterior aprendimos los fundamentos para trabajar con *Milkshape3D*. Continuamos la lección realizando algunos ejercicios para practicar el modelado en baja a partir de primitivas y mediante el sistema de modelado por vértices. La aplicación se encuentra en el CD-ROM.

MODELANDO EL CUERPO DE UN AUTOMÓVIL POR VÉRTICES

El sistema de modelado por vértices es algo lento y tedioso, pero se puede construir cualquier cosa y controlar mejor el número de polígonos. Vamos a tomar contacto modelando -de una manera básica- el casco de un automóvil.

En primer lugar, realizaremos la silueta lateral del casco colocando vértices en la vista lateral derecha como se muestra en la figura 1. Para situar vértices pulsamos la opción "Vertex" del panel de herramientas.

Una vez dibujada la silueta, procederemos a unir cada vértice entre sí para formas las caras.

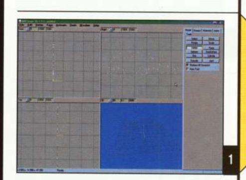
Pulsamos en "Face" en el panel de herramientas, y pulsamos en tres vértices consecutivos según el orden que mostramos en la figura 2 para ir formando los triángulos. En ocasiones puede ocurrir que nos equivoquemos en la secuencia; entonces podemos ver en la vista 3D cómo la cara se dibuja de color oscuro. Esto significa que el orden de las caras está al revés. Una solución es elegir la opción "Reverse Vertex Order" en el menú "Face" o bien deshacer con *CTRL + Z* y volver a construir la cara. Aconsejamos esto último, para ir adquiriendo soltura.

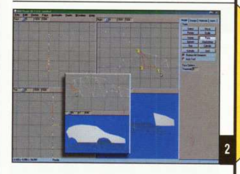
Una vez formada la silueta, observamos que nuestro modelo es plano, debido a que hemos trabajado en una de las vistas. Para dar forma en 3D, vamos a realizar un *extrude* de todas las caras hacia la derecha. Elegimos "Select" (F1) y "Face" en opciones, y con un recuadro seleccionamos todas las caras en la vista lateral. A continuación elegimos la herramienta "Extrude" y nos situamos en la vista frontal para desplazar el ratón hacia la derecha con el botón izquierdo pulsado, como vemos en la figura 3.

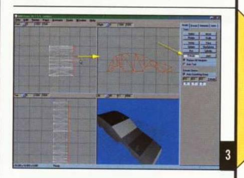
Ya tenemos el cuerpo de medio vehículo. Para la otra mitad sólo tenemos que seleccionar todo, duplicar y realizar un espejo lateral de la copia; es decir, colocar la mitad copia a continuación de la original pero con simetría inversa. Para realizar las simetrías tenemos las opciones de "Mirror" situadas en el menú "Vertex". Es importante colocar el extremo interior de la pieza original a duplicar, alineado con los ejes, para que la operación de simetría se realice con precisión, como en la figura 4. Muy bien. Duplicamos con *CTRL +D* y realizamos una simetría de izquierda a derecha con "Mirror Left <— > Right". Ahora nos queda unir las dos partes. Seleccionamos los vértices centrales con "Ignore BackFaces" deseleccionado y elegimos la opción "Weld Together" en el menú

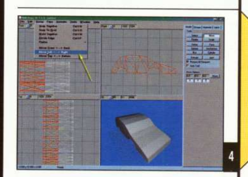
TRUCO

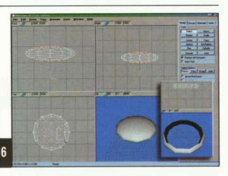
Para cambiar el orden de los vértices podemos elegir la opción "Reverse Vertex Order" en el menú "Face".


1

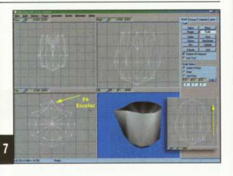

2

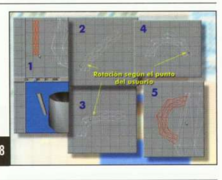

3


4

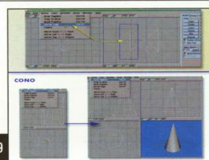


5


6



7



8

9

"Vertex"o *CTRL + W* para realizar la unión. Pero todavía seguimos teniendo dos objetos diferentes, aunque unidos entre sí. Para convertirlos en un solo modelo debemos reagruparlos. Elegimos la opción "Group" de la herramienta "Select" (F1) y seleccionamos las dos partes. Pulsamos en la pestaña "Group" de la ventana de herramientas y pulsamos sobre "Regroup". Esta operación une las dos partes, reagrupándolas en una, tal y como vemos en la figura 5.

OPERACIONES CON PRIMITIVAS

Para terminar nuestra preparación, vamos a modelar pequeños objetos a partir de primitivas para poder practicar todas las herramientas y funciones de modelado.

En primer lugar, vamos a construir un anillo a partir de una esfera de 12 partes (*stacks*) y 12 divisiones (*slices*). Pulsamos la herramienta "Sphere", y en "Sphere Options" elegimos el tamaño. Una vez dibujada, pasaremos a realizar un escalado en su eje Y para achatarla. Pulsamos en "Scale" (F4) y en la casilla Y de opciones escribimos el valor 0.3 y pulsamos en "Scale". Esta forma es más exacta que escalar la esfera manualmente desde cualquiera de las vistas. Para el agujero del centro, nos situamos en la vista superior (Top) y seleccionamos los vértices centrales poco a poco (utilizar la tecla *SHIFT* para sumar selección) con "Ignore BackFaces" desactivado, como vemos en la figura 6. A continuación borramos (*SUPR*) y ya tenemos el agujero.

Vamos a continuar practicando, modelando una jarra.

TRUCO

Podemos sumar selecciones, manteniendo la tecla SHIFT pulsada mientras seleccionamos.

Partimos de una esfera de 4 stacks y 12 slices (4 x 12). Seleccionamos los vértices superiores y los desplazamos en la vista lateral hacia arriba. Luego, desplazamos hacia abajo el vértice de la unión superior para simular un hueco interior. Vemos en la figura 7 cómo para la boquilla seleccionamos tres vértices de uno de los bordes en la vista lateral y en la vista superior los escalamos (F4) hasta dar la forma. El asa la haremos separadamente y luego la colocaremos en la parte trasera del cuerpo.

Partimos de un cilindro de 4 x 6. Para dar la forma curva debemos seleccionar los vértices de cada parte del cilindro y rotar según el punto del usuario, como se muestra en la figura 8. Una vez curvado, seleccionamos toda el asa y la rotamos y movemos hasta el cuerpo de la jarra.

A veces ocurre que necesitamos tener más vértices de la cuenta en una primitiva, para poder modificarla y obtener formas más complejas. Para solucionar este problema existe una función que divide dos vértices seleccionados, colocando otro en medio, como vemos en la figura 9.

Dibujemos un cubo. Seleccionamos dos de sus vértices y elegimos la opción "Divide Edge" (*CTRL + P*) del menú "Vertex". Vemos cómo aparece un vértice entre los dos seleccionados y se ha formado otro triángulo nuevo.

Para terminar, vamos a crear un cono a partir de un cilindro para aprender a utilizar la función destinada a unir varios vértices en uno.

Dibujamos un cilindro de 1 x 8 y seleccionamos todos los vértices superiores. A continuación elegimos la opción "Snap Together" (*CTRL + P*) en el menú "Vertex".

La mejor manera de dominar *Milkshape3D* es practicando mucho. En el próximo número haremos una pausa para dedicarnos al grafismo 2D de lleno, porque vamos a empezar a trabajar en nuestro juego.

MODELADO 2D

HERRAMIENTAS DE PAINT SHOP PRO 7

En la entrega anterior, tuvimos nuestra primera toma de contacto con *Paint Shop Pro* (incluido en el CD-ROM). A continuación vamos a conocer la paleta de herramientas y todas sus opciones.

En la "paleta de herramientas" se encuentran todas las herramientas necesarias para pintar, dibujar, seleccionar y retocar imágenes. Asociada a cada herramienta, aparecen sus opciones correspondientes en la ventana "Tool Options". Para empezar vamos a crear una imagen nueva (*CTRL + N*) de 400 x 400 píxels a 72 (píxel/pulgada) de resolución y con fondo negro. Vamos a rellenar la nueva imagen con una textura.

Seleccionamos la herramienta de relleno y en la paleta de colores "Colors" elegimos la segunda textura de la lista en "Styles" (ver fig. 10). Y hacemos clic sobre la imagen.

La herramienta nos permitirá acercarnos o alejarnos en la imagen completa o en el trozo que seleccionemos. (Observad los factores de zoom en la ventana de opciones).

La siguiente herramienta nos permitirá recortar la imagen en el tamaño que hayamos seleccionado. Para activar el recorte debemos pulsar en el botón "Crop Image" de la paleta de opciones. Después de recortar podemos volver a la imagen original haciendo un *undo* (*CTRL + Z*).

Tenemos varias opciones de selección geométrica. Desde el típico cuadrado hasta selección en forma de estrella. Además, podemos elegir una selección con bordes más o menos finos (Feather) y disponer de suavizado de bordes (Antialias).

Asimismo, con la herramienta podemos realizar selecciones a mano alzada, punto a punto, o seleccionar bordes entre dos áreas de distinto contraste de color o iluminación (Smart Edge). (ver fig. 11).

Practica con la imagen estas herramientas. Recuerda que para cerrar una selección basta con hacer dos *clics* seguidos. Observa también que, una vez hecha la selección, el cursor se cambia por el de "mover" cuando estás encima del área seleccionada. Esto significa que si desplazas la selección cortarás el fondo y moverás el área seleccionada.

Paint Shop Pro permite hacer copias (*CTRL + C*) de una selección en otra capa (*CTRL + L*), crear una imagen nueva con sólo el área seleccionada (*CTRL + V*), crear una copia de la selección en la misma imagen (*CTRL + E*) o crear una selección transparente (*CTRL + SHIFT + E*).

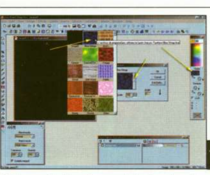
Otra herramienta de selección muy útil es la varita mágica.

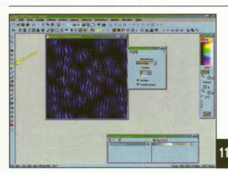
Este tipo de selección permite escoger distintas áreas del dibujo basándose en 5 tipos de valores: por cantidad de color rojo, verde o azul (RGB Value); por tono de color (Hue); por luminosidad o cantidad de blanco (Brightness); selección de áreas no transparentes (All Opaque); y por la opacidad de los píxels (Opacity). También permite controlar la similitud entre píxels (Tolerance) y el grosor del borde de selección (Feather).

La opción "Sample Merged" activada indica que se hará la selección en todas las capas, mientras que si está inactiva lo hará sólo en la capa activa.

Esta herramienta ("cuentagotas") sirve para seleccionar un color en la imagen. En herramientas como "pincel" o "bote de relleno", podemos activar momentáneamente el cuentagotas manteniendo pulsada la tecla *CTRL*.

Antes de ver las herramientas para pintar, vamos a practi-


10
Podemos seleccionar la segunda textura deseada.

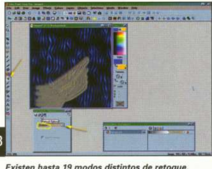

11
Podemos realizar distintos tipos muy precisos de selecciones.

TRUCO

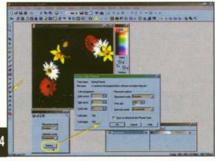
Para trabajar mejor, podemos ocultar la paleta de herramientas pulsando la tecla "P" y seleccionar las herramientas desde la ventana "Tool Options" pulsando en el icono .

NOTA

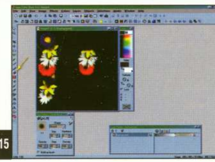
***Paint Shop Pro* posee un historial de "hacer y deshacer" situado en el menú "Edit" llamado "Command History" (CTRL + SHIFT + Z), que nos permite ir hacia atrás o hacia delante en todas las acciones que hayamos realizado.**



Conmutamos el color de fondo por el de tinta.



Existen hasta 19 modos distintos de retoque.



Gracias a las opciones, podemos controlar cómo pintar las imágenes.

15

Dibujamos una fila de flores.

car con dos herramientas muy interesantes: reemplazar color y retocar. La primera nos permite reemplazar un color de la imagen que elijamos por otro color. Pulsamos sobre el icono, con *CTRL* pulsado elegimos con el cuentagotas el color que queremos sustituir. A continuación, en la paleta de colores, colocamos en el color de fondo el que reemplazará al elegido anteriormente. Conmutamos el color de fondo por el de tinta y pintamos (ver fig. 12). Para sustituir de nuevo el color, basta con conmutar de nuevo y pintar.

La siguiente herramienta, "Retouch", realiza cambios en la imagen según vamos pintando. En el panel de opciones en la pestaña "Retouch Options" podemos elegir hasta 19 modos de retoque. (ver fig. 13).

Vamos a continuar viendo las herramientas de pintar: pincel, clonado, pincel con forma, y spray.

Antes de nada vamos a borrar toda la imagen. Seleccionamos todo con *CTRL + A* y borramos con la tecla *Supr (DEL)*. La pantalla se quedará con el color de fondo seleccionado.

La herramienta nos servirá para dibujar como si estuviéramos pintando con un pincel. En el panel de opciones podemos modificar todos las características de la brocha como tamaño, forma, cantidad de pintura o transparencia.

Con esta herramienta podemos pintar con el dibujo que queramos. Al seleccionarla, nos aparece en la paleta de opciones una bola. Pulsando en la pequeña flechita de la derecha accedemos a la librería de imágenes de *Paint Shop Pro*. En la pestaña de opciones, podemos controlar cómo se pintarán esas imágenes (ver fig. 14).

Para probar la herramienta de clonado, vamos a dibujar primero una fila de flores (ver fig. 15).

Seleccionamos la herramienta de clonado. Marcamos con el botón derecho del ratón sobre lo que queremos clonar. Una vez hecho esto podemos pintar en otro sitio de la imagen, en donde se irá copiando a partir del punto que marcamos al principio.



TRUCO

Para deseleccionar una selección en cualquier momento pulsar CTRL + D). Otra herramienta de selección muy útil es la varita mágica.



TRUCO

Para cambiar el color de fondo por el de tinta en la paleta de colores hay que pulsar en las flechas que están entre ambos.



TRUCO

Para aumentar o reducir el tamaño de visualización de una imagen, basta con pulsar las teclas + o –.



TRUCO

Si queremos tener nuestras propias imágenes para pintar con ellas, tenemos que copiar nuestros ficheros de imágenes en el directorio que tiene Paint Shop Pro para ello.



En el próximo número...

... nos quedan por ver las herramientas de texto, dibujo y polígonos, pero lo dejaremos para el próximo número; vamos a utilizarlas en el diseño del logotipo de *Zone of Fighters*.

El programa Goldwave

En el número anterior vimos las herramientas que utilizaremos para desarrollar el sonido de nuestro juego. En estas páginas, vamos a conocer un estupendo y asequible editor de sonido: *GoldWave* (incluido en el CD-ROM).

Con esta aplicación podemos grabar, editar y convertir sonidos, además de extraer en ficheros de audio las pistas de un CD. Puede tener muchos sonidos abiertos a la vez, con la posibilidad de copiar partes de audio de una ventana a otra. Puede convertir a multitud de formatos de audio incluido MP3.

CÓMO EMPEZAR

Al ejecutar *GoldWave* observamos una ventana flotante de control. Aquí podemos controlar la ejecución y grabación de sonido, su volumen, balance y velocidad de reproducción. Asimismo, disponemos de un gráfico del aspecto del sonido en tiempo real. En el botón de propiedades, podemos configurar las opciones de reproducción, grabación, volumen de dispositivos, gráficos y dispositivos instalados.

Vamos a crear un plantilla de sonido nueva. Para ello, pulsamos en el botón "New". Aparecerá una ventana de diálogo para elegir las características del sonido. Vamos a elegir calidad "CD" y pulsamos "OK". Se abre una ventana de sonido. Si pulsamos de nuevo en "New" se abrirá otra nueva ventana de sonido y así todas las que nuestro sistema permita (ver fig. 1).

Para conocer las herramientas de este programa, vamos a extraer una pista de un CD de música.

Cerramos todas las ventanas anteriores. Introducimos un CD de música y elegimos la opción "CD audio extraction" del menú "Tools". Elegimos nuestro CD ROM y la pista que queremos extraer en la ventana "From". A continuación (para no extraer toda la pista) en "To" seleccionamos el final de la pista para reducir la longitud del fichero (unos 15 segundos). Pulsamos en "Save" y saldrá una ventana que nos pide dónde guardaremos el audio extraído. Escribimos "prueba" y "Guardar". (La extracción comenzará) (ver fig. 2).

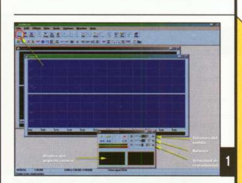
Bien, ya tenemos nuestros 15 segundos de música en la ventana de audio. Pulsad para reproducirla y probad a manejar los controles en la ventana de control. (Ver figura controles).

EDITAR UN SONIDO

Vamos a aprender seguidamente a seleccionar, cortar, copiar, insertar y manejar la ventana de audio.

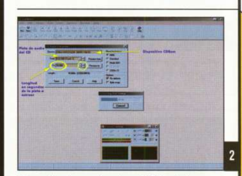
En primer lugar, para ver toda la onda de sonido en la ventana de audio pulsamos en el icono para ajustar la vista.

Para borrar un trozo del sonido, por ejemplo, el silencio del principio de la onda, debemos seleccionar el trozo primero. Moviendo el ratón mientras hacemos *clic* con el botón izquierdo, desplazamos la marca de comienzo de selección y si lo hacemos con el botón derecho moveremos la marca de final de selección. Seleccionemos pues el comienzo de la onda sin sonido. Nos acercamos a la selección para ajustarla correctamente pulsando en el icono. Desplazamos la barra deslizado-



1

Podemos abrir todas las ventanas de sonido que queramos.



2

Extracción de una pista de un CD de música.

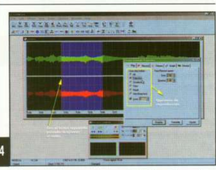


NOTA

Si el sonido es estéreo, la ventana se divide en dos partes. La parte superior corresponde al canal izquierdo y la inferior al derecho. Para conmutar cualquiera de los dos canales hay que hacer clic en el icono.

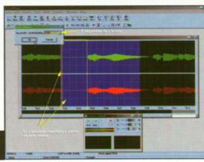
3

Podemos editar el sonido y cortar los silencios a nuestro antojo.



4

Seleccionando una parte central del sonido podremos probar las opciones.



5

De este modo insertamos un silencio en el sonido.

ra situada en la parte inferior de la ventana de audio hasta ver el final de la selección y con el botón derecho pulsado, movemos el ratón hasta ajustar la marca del final. Ya estamos preparados para cortar el silencio. Pulsamos en el icono o con *CRTL + X*. Volvemos a ajustar toda la onda a la ventana y listo (ver fig. 3).

Podemos también seleccionar todo el sonido de una vez con *CTRL + A* y escoger sólo el que vemos en la ventana con *CTRL + W*.

Vamos a aclarar cómo funcionan los controles de "Play". El primer control sólo reproduce lo que está seleccionado una sola vez. Si queremos un mayor control de reproducción tenemos que elegir el icono . En la pestaña "Play" de opciones encontramos todas las posibilidades de reproducción en "User play button". Antes de nada, seleccionemos una parte central del sonido para probar todas estas opciones (ver fig. 4).

- **1. "All".** Reproduce todo desde el principio.
- **2. "Selection".** Reproduce toda la selección
- **3. "Unselected".** Reproduce todo lo que no está seleccionado. Cuando llega a la marca del comienzo salta hasta la marca del final y continúa.
- **4. "View".** Reproduce lo que se ve en la ventana.
- **5. "Finish".** Reproduce tres segundos antes de llegar a la marca final.
- **6. "Intro/Loop/end".** Se reproduce desde el principio, hace un bucle de la selección y sigue hasta el final.
- **7. "Loop".** Si está marcado, repetirá cuantas veces se indique una reproducción de la selección.

Vamos a seguir copiando una selección e insertándola en cualquier parte del sonido.

Primero, hagamos una selección. Para copiar en memoria pulsamos el icono o *CTRL + C*. Para insertar en el comienzo del sonido elegir la opción "Paste at" Beginning" en el menú "Edit" o *CTRL + B*. Para insertar al final *CTRL + F* ("Paste at End") y para insertar en el interior del sonido *CTRL + F* ("Finish market").

También podemos realizar una copia de la memoria, pegando en una nueva ventana, creando así un nuevo sonido. Esta operación la realizamos con la opción "Paste new" (*CTRL + P*) en el menú "Edit".

Hay dos opciones muy interesantes en el menú de edición que son "Mix" y "Replace".

La opción "Mix" (*CTRL + M*) permite mezclar la copia contenida en memoria con la selección actual. Antes de mezclar, proporciona la opción de hacerlo con más o menos volumen.

La siguiente herramienta es "Replace", que admite reemplazar el trozo de sonido copiado en memoria por el trozo seleccionado en el sonido.

Por último, la opción "Trim"(*CTRL + T*) del menú "Edit" es similar a la función de "recortar" utilizada en diseño gráfico, por ejemplo, por el *Paint Shop Pro*. Borrará todo el sonido que no esté seleccionado, y lo que queda se ajusta a la ventana.

Para finalizar, vamos a insertar un silencio en el sonido de 2 segundos. Este espacio sin audio se insertará después de la marca de comienzo de selección. Colocamos la marca donde queremos hacer el inserto. Elegimos la opción "Insert silence" del menú "Edit" y en la casilla numérica escribimos 2.0 y "OK" (ver fig. 5).

Todavía nos queda mucho que aprender de este programa, pero lo haremos como siempre, en la práctica. Así que ...

TRUCO

Si hacemos clic en cualquiera de las ventanas del gráfico del sonido mientras reproducimos, cambiará la forma de representación.

En el próximo número...

... empezaremos a realizar los efectos especiales de audio de nuestro juego.

Instrucciones básicas del lenguaje

En el número anterior vimos algunos conceptos básicos de programación, necesarios para entender la filosofía del lenguaje *BASIC* que se esconde tras *Blitz3D* (B3D)

SENTENCIAS

Antes de continuar, es necesario conocer qué son las sentencias. En realidad, cuando ejecutamos un programa, le estamos dando al ordenador un conjunto de órdenes definidas. Estas órdenes se denominan sentencias y las hay de distinto tipo según su funcionalidad:

- 1. Sentencias de flujo de programa, comúnmente llamadas "sentencias condicionales".
- 2. Sentencias para bucles.
- 3. Sentencias para cálculos o de asignación.
- 4. Sentencias de control de programa.

FLUJO DEL PROGRAMA

El flujo de un programa es controlado por sentencias condicionales, las cuales permiten cambiar el rumbo de la ejecución del programa después de comprobar y evaluar si se cumple cierta condición. Estas condiciones están formadas por expresiones que contienen operadores de comparación, por ejemplo, una expresión de este tipo podría ser *vida < 1* o *x=100 And y=200*

Los símbolos "<" o "And" son los operadores de comparación y lógicos, y en *B3D* disponemos de los siguientes: < (menor que), > (mayor que), <= (menor o igual que), >= (mayor o igual que), = (igual que), <> (distinto que). En todos estos operandos se necesitan dos valores para comparar. También disponemos de operandos lógicos como: *Not* (compara que no se cumpla una condición), *And* (compara que dos expresiones son ciertas), *Or* (compara que al menos una de las dos expresiones es cierta), *Xor* (compara que sólo una de las dos expresiones es cierta).

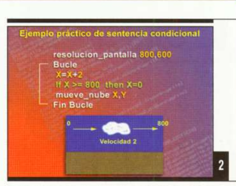
Siempre que realicemos estas comprobaciones, obtendremos como resultado los valores 0 (False) o 1 (True). Además, podemos encadenar expresiones condicionales separadas por operadores lógicos y englobar expresiones dentro de otras utilizando paréntesis, por ejemplo:

```
(vida<1 or energia<100)
and puntos=0.
```

En este ejemplo, se evalúa primero lo que está dentro del paréntesis y el resultado se comparará con la siguientes expresión; así, se cumpliría la condición *si puntos vale 0* y *vida es menor que 1 o si puntos vale 0* y *energía es menor de 100*.



En la figura se muestra una tabla con los tipos de operadores lógicos y condicionales disponibles en Blitz3D.

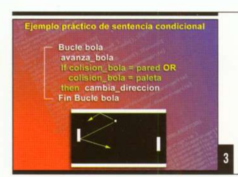


Un esquema del movimiento continuo y circular de una nube en el horizonte. Se actualizan coordenadas por medio de una condición.

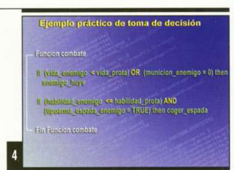
Hay que tener en cuenta que los operadores lógicos convierten sus operandos a valores enteros y el resultado que producen también. A continuación, se muestran algunos ejemplos de expresiones condicionales válidas:

```
coordx_enemigo <> coordx_jugador
And Not disparo=0
```

En esta expresión se comprueba si las coordenadas x del enemigo y del jugador son diferentes y que, por ejemplo, se haya producido un disparo, es decir, negar el valor 0 de disparo es ponerlo a 1; luego la expresión también podría escribirse como *And disparo=1*.



Un ejemplo del uso de un operador lógico OR para el control del rebote de una pelota.



La combinación de operadores lógicos y de comparación permiten simular cierta inteligencia en un programa.

Con este tipo de sentencias es como hacemos que nuestro programa "piense" y tome decisiones.

Pero para implementar estos controles es preciso disponer de estructuras establecidas para ordenar el flujo de un programa. *B3D* posee varios tipos de estas estructuras condicionales.

ESTRUCTURA IF ... THEN

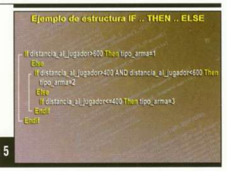
Su construcción básica es la siguiente:

```
If expresión Then sentencias1
Else sentencias2
```

o también

```
If expresión
  Sentencias1
Else
  Sentencias2
Endif
```

Se evaluará la expresión y si (If) el resultado es verdadero entonces (Then) se ejecutarán las sentencias1; si el resultado es falso, se ejecutarán las sentencias2. Por ejemplo:



Las estructuras IF .. THEN se pueden combinar entre sí.

```
If vida<1 Then jugador_muere
Else vida=vida-1
```

Si (If) *vida* es menor que 1 entonces (Then) se ejecuta la función "jugador_muere"; si no (Else), se decrementa una vida.

Se pueden encadenar varias estructuras *If.*

```
If expresión1
  Sentencias1
Else If expresion2
      Sentencias2
     Else
      Sentencias3
Endif
```

En esta estructura encadenada, las "Sentencias3" se ejecutarán solamente si la "expresion1" del *If* o la "expresion2" del *Else If* (ElseIf) fueran verdaderas (True)

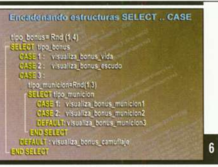
ESTRUCTURA SELECT ... CASE

```
Select expresión
  Case expresiones1
  Case expresiones2
  Default expresiones3
End Select
```

Este tipo de estructuras nos ayuda a evitar un gran número de estructuras If ... Then en nuestro programa. Imaginaos que queréis dibujar en pantalla diferentes explosiones dependiendo del tipo de disparo que se haya efectuado. Esto se podría realizar mediante muchos *If* , por ejemplo,

```
If disparo=1 Then
dibujar_explosion1
If disparo=2 Then
dibujar_explosion2
If disparo=3 Then
dibujar_explosion3
```

Realmente este sistema es algo engorroso y poco elegante. La estructura "Select ... Case" es una alternativa más apropiada.



Las estructuras SELECT.. CASE son fácilmente encadenadas.

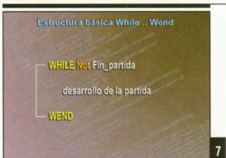
```
Select disparo; según el valor
de disparo
Case 1 Dibujar_explosion1 ;
  en caso de que disparo=1
  dibujamos la explosión 1
Case 2 Dibujar_explosion2 ;
  en caso de que disparo=2
  dibujamos la explosión 2
Default Dibujar_explosion3 ;
  en el caso restante, es
  decir, disparo=3
  dibujamos la explosión 3
End Case
```

SENTENCIAS PARA BUCLES

En este tipo de estructuras se realiza un bucle de las sentencias hasta cumplirse determinada condición.

ESTRUCTURA WHILE ... WEND

```
While expresion
  sentencias
```



Una estructura WHILE ..WEND permite controlar el flujo de la partida.

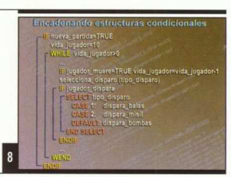
```
Wend
```

Mientras (While) se cumpla la expresión, se ejecutan las sentencias.

Esta estructura es utilizada cuando queremos que se realicen operaciones sólo cuando se cumpla cierta condición. Antes de entrar en la estructura para realizar el grupo de sentencias, se evalúa la expresión y mientras ésta sea verdadera, se permanecerá en el interior de la estructura. En el ejemplo:

```
vida_jugador=3
While vida_jugador>0
 dibuja y mueve jugador
 dibuja y mueve los enemigos
 ataca los enemigos al jugador
 If disparo_enemigo colisiona
 con jugador Then
 vida_jugador=vida_jugador-1
Wend
muere_jugador
```

Asignamos 3 vidas al jugador. Mientras (While) las "vidas del jugador" no sean 0, "dibujaremos y moveremos al jugador y a los enemigos", "atacarán los enemigos". Si un "disparo del enemigo" "alcanza al jugador", éste pierde una vida. Se vuelve a preguntar si la "vida del jugador" no es 0; como ahora vale 2 se vuelve a ejecutar las acciones anteriores y así hasta que las vidas del jugador valgan 0. Se saldrá de la estructura y se ejecutará "muere_jugador".



Se pueden encadenar distintos tipos de estructuras condicionales, según las necesidades del programa.

ESTRUCTURA REPEAT ... UNTIL

```
Repeat
  sentencias
Until expresion
```

Repetir (Repeat) las sentencias hasta (Until) que se cumpla la expresión.

Su misión y funcionamiento es la misma que la estructura "While ...End". La única diferencia es que la evaluación de la expresión se realiza después de ejecutar las sentencias contenidas en la estructura. Se utiliza cuando queremos que el grupo de sentencias se ejecute al menos una sola vez. Vamos a realizar el ejemplo anterior pero con esta estructura:

```
vida_jugador=3
Repeat
 dibuja y mueve jugador
 dibuja y mueve los enemigos
 ataca los enemigos al jugador
 If disparo_enemigo colisiona
 con jugador Then
 vida_jugador=vida_jugador-1
Until vida_jugador=0
muere_jugador
```

ESTRUCTURA REPEAT ... FOREVER

```
Repeat
  Sentencias
Forever
```

Repetir (Repeat) las sentencias para siempre (Forever).

En realidad esta estructura sirve para realizar un bucle infinito.

Si en algún momento deseamos salir de este bucle debemos utilizar la función Exit. Por ejemplo:

```
Repeat
 Print "esto no tiene fin
 hasta que aprietes un tecla"
 If aprietas cualquier tecla
 Then Exit
Forever
```

El mismo resultado con dos estructuras de bucles distintas.

ESTRUCTURA FOR ... NEXT

```
For variable = valor inicial
To valor final Step salto en
el incremento del valor
 sentencias
Next
```

Este tipo de estructuras para bucles funciona preguntando por la evolución del valor de una variable. Antes de entrar en el bucle, se asigna el "valor inicial" a la "variable" y se entra en el bucle. Una vez ejecutadas las "sentencias" con la función "Next" se regresa al *For*. Se incrementa 1 unidad el valor de "variable" (opcionalmente se puede incrementar en las unidades que se indique en "Step") y se vuelve a entrar en el bucle. Esta operación se repetirá hasta que el valor contenido en "variable" alcance el "valor final". Por ejemplo, queremos contar del 1 al 10 de dos en dos e imprimirlo en la pantalla:



En la figura, el bucle interior (1) se ejecuta antes que el exterior (2).

Estructura For .. Next frente a Repeat ..Until

```
FOR x = 1 TO 800
   Mover_nave x,100
NEXT

x = 0
REPEAT
   x = x + 1
   Mover_nave x,100
UNTIL x=800
```

El mismo bucle con dos estructuras diferentes.

```
For contador = 1 to 10 step 2
  print contador
Next
```

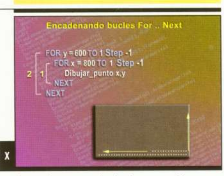
Para realizar la cuenta pero al revés, se deben cambiar el "valor final" por el "valor inicial" y colocar "Step" en negativo. Por ejemplo, si queremos hacer una cuenta atrás del 10 al 1:

```
For contador =10 to 1 Step -1
  print contador
Next
```

También puede haber bucles dentro de otros bucles. El bucle interior se repetirá tantas veces como indique el bucle exterior.

En *B3D* existe un tipo de bucle *For* que nos permite recorrer cada uno de los objetos que forman parte de una estructura de datos. Se trata de la estructura For ... Each

```
For variable = Each nombre de
la estructura de datos o tipo
de datos
```



Se pueden recorrer los bucles en sentido contrario con la opción STEP.

```
  Sentencias
Next
```

Es muy utilizado para actualizar grandes grupos de objetos. Pero explicaremos el uso de esta sentencia con más detalle cuando hablemos de las estructuras de datos.

USO DE MATRICES

En el número anterior explicamos qué eran las matrices o *arrays*. Ahora le daremos algún uso en nuestro programa. Supongamos que queremos guardar información sobre la posición de cuatro llaves dentro de un laberinto en 2D. Estas posiciones vienen determinadas por las coordenadas X e Y. Definimos la matriz: Dim llave (3,1) ="10,20, 20,30, 150,340, 100,400"

La primera llave (la número 0) está en las coordenadas X=10 e Y=20, la segunda en X=20 e Y=30, y así sucesivamente. En el siguiente ejemplo imprimimos la coordenada Y de la llave 3.

```
Dim llave (3,1) ="10,20,
20,30, 150,340, 100,400"
Print llave(3,1)
```

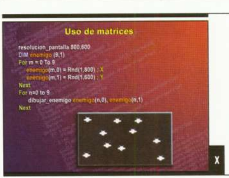
Y en el siguiente, imprimimos las coordenadas X e Y de las cuatro llaves:

```
Dim llave (3,1) ="10,20,
20,30, 150,340, 100,400"
```

Damos valores a la matriz

```
For numero_llave=0 To 3;
Recorremos la tabla
 Print llave (numero_llave,0);
 imprimimos la coordenada X de
 la posición numero_llave
 Print llave (numero_llave,1);
 imprimimos la coordenada Y de
 la posición numero_llave
Next
```

Las matrices pueden resultar útiles para manejar datos con un elemento común, pero si tenemos un grupo de objetos con los mismos parámetros no resulta del todo eficaz.



Un ejemplo de uso de matrices para imprimir un grupo de ovnis.

Supongamos que queremos controlar la posición (Coordenadas X e Y) y los gráficos (3 tipos de gráficos) de una horda de 50 naves enemigas. Se podría utilizar una matriz como Dim enemigos (49,4) y luego recorrerla con un bucle *For ... Next*. Esto requiere mucho esfuerzo por parte del ordenador y del programador. Para solucionar este problema *B3D* ofrece la posibilidad de definir tipos de datos en estructuras llamadas *Type*. El ejemplo anterior se podría definir con este tipo de estructura de la siguiente forma:

```
Type tipo_enemigo;
Definimos un cómo serán los
enemigos.
 Field x,y;
Coordenadas X e Y de cada enemigo
 Field grafico[2];
3 gráficos diferentes por
cada uno de ellos
End Type
Dim enemigos.tipo_enemigo(49);
Creamos 50 enemigos distintos
de la misma clase
For n=0 to 49
 enemigos(n)= New tipo_enemigo;
Inicializamos lo 50 nuevos
enemigos para poder usarlos
Next
```

Realmente, es algo difícil de entender, pero más rápido, eficaz y elegante. Cuando comprendas su funcionamiento, seguramente utilizarás este sistema las veces que puedas en tus programas. No te preocupes, en el siguiente número explicaremos con más detalles este fantástico sistema.

Secuenciadores. Cubase VST (I)

La casa Steinberg se remonta a la época del Atari ST. Desde entonces, ha conservado su filosofía de diseño. Tanto que, si has conocido la primera versión de su secuenciador *Cubase*, no tendrás problema para adaptarte a la última versión, casi veinte años después. (Incluido en el CD-ROM).

La saga de secuenciadores *Cubase* ha sido su producto estrella y ha creado un estándar mundial en la informática musical doméstica y profesional.

Con la llegada de las posibilidades multimedia, Steinberg adaptó su secuenciador *Cubase* a los nuevos tiempos, incluyendo en su gama VST la posibilidad de mezclar pistas de audio con las tradicionales pistas MIDI. Actualmente, es utilizado en estudios de grabación profesionales, en cine, y en cualquier medio audiovisual, y por supuesto por miles de músicos.

LOS PRIMEROS PASOS

Para empezar, debemos saber que *Cubase* es, ante todo, un secuenciador MIDI. No obstante, posee la capacidad de grabar y reproducir pistas de audio y mezclarlas. Para algunos aspectos de este tutorial, sería conveniente que tuvieras conectado al ordenador un instrumento MIDI, como un teclado maestro. *Cubase* es un programa muy extenso y potente, así que en estos tutoriales expondremos una manera rápida y pragmática de utilizarlo. Lo que aprendas aquí puedes aplicarlo a cualquier versión de la gama VST de *Cubase*.



Configuración por defecto al arrancar Cubase.

Al ejecutar *Cubase* se carga una configuración básica por defecto, consistente en una ventana de arreglo con 8 pistas de audio y 16 pistas MIDI (ver fig. 1).

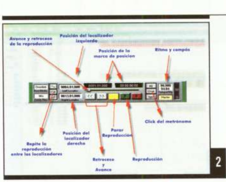
En esta primera entrega conoceremos el uso de los elementos de la interfaz. Para ello, vamos a cargar la canción "def.all" que se encuentra en el directorio "\audio" del CD con la función "Open" del menú "File". Esta canción es un pequeño ejemplo que contiene una pista de audio y ocho pistas MIDI.

Para empezar, presentaremos un elemento muy importante en el *Cubase*: la barra de transporte. Es un panel flotante que contiene las funciones de reproducción, paro, rebobinado, grabación, ajuste de ritmo, etc. En la figura 1 se puede ver un esquema de este panel y una descripción de sus funciones. Es importante que aprendas la equivalencia de los botones de la barra de transporte en el teclado, ya que usar el ratón cuando grabas o manipulas esta paleta es un poco engorroso.

LA BARRA DE TRANSPORTE

Vamos a practicar la reproducción. Sobre los botones de control podemos ver una barra deslizadora que nos permitirá desplazar la marca de posición de la canción igual que utilizamos los botones de "Atrás" y "Adelante" (ver fig. 2).

Iniciamos la reproducción pulsando el botón "Play" o con la



Con la barra de transporte reproducimos las canciones.

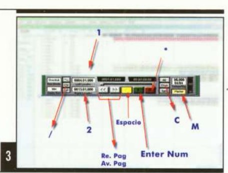
tecla "*Intro*" del teclado numérico. Para parar la ejecución pulsamos el botón "Stop" o la barra espaciadora. Para irnos a la posición del localizador izquierdo pulsamos "1" (por defecto, siempre estaremos hablando del teclado numérico, mientras no se indique lo contrario) y para colocarnos en la posición del localizador derecho pulsamos "2". Si lo que deseamos es ir al comienzo de la can-

NOTA

Básicamente, sólo existen 16 canales MIDI distintos. Esto no significa que estemos limitados a 16 pistas MIDI; es decir, podemos tener 10 pistas diferentes asignadas al canal MIDI 5. Las 10 pistas sonarán con el mismo sonido pero pueden tener distintos arreglos musicales.

TRUCO

Se puede utilizar el teclado numérico para reemplazar a la barra de transporte en cualquier momento, incluso estando en otra ventana de diálogo.



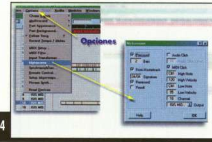
Podemos avanzar o retroceder a nuestro antojo.

ción pulsamos "5" o la tecla "*Inicio*". Para ir al último sitio donde se paró la música pulsamos el "9".

Para correr hacia delante o hacia atrás utilizaremos las teclas retroceder página *RePag* y avanzar página *AvPag* situadas fuera del teclado numérico (ver fig. 3).

A continuación, vamos a controlar la posición de los localizadores. Estos elementos nos sirven para indicar el comienzo y el final de la grabación o de la reproducción cíclica. Para colocar o desplazar el localizador izquierdo, nos situamos en la regla de compases y pulsamos con el botón izquierdo del ratón donde queremos colocarlo; realizaremos lo mismo, pero con el botón derecho, para el localizador derecho. Podemos realizar la misma operación introduciendo los valores en las casillas correspondientes en la barra de transporte, haciendo *clic* dos veces.

Otras funciones de la barra de transporte son: el *clic* de metrónomo, el sincronismo, el ritmo y unidad de tiempo, la reproducción cíclica, el modo de grabación y el tipo de grabación cíclica. No las veremos ahora todas pero sí las más utilizadas.



El metrónomo es una función que nos será de gran utilidad.

Una función muy útil es el metrónomo, podemos activarlo pulsando la tecla *C* y configurarlo en la opción "Metronome" en el menú "Options" (ver fig. 4).

Si queremos que se repita la reproducción entre los localizadores, debemos activar la "Reproducción cíclica" con la tecla "*/*".

Mediante el botón "Master" (tecla "*M*") activamos el tiempo y el compás definidos en lista del "Maestro de pistas" (Mastertracks), que explicaremos en otra ocasión. Si lo tenemos desactivado cambiarán a los valores por defecto.

LA VENTANA DE ARREGLO

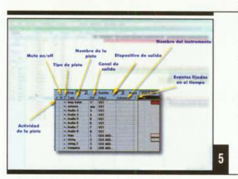
El arreglo está formado por pistas, donde grabamos y reproducimos la canción. El sistema de pistas permite grabar los instrumentos por separado para poder editarlos uno a uno. En la figura 2 se puede ver una descripción de la parte de datos de la ventana de arreglo y en la figura 3 una descripción del visualizador del arreglo. Aquí es donde vemos la música grabada en cada pista y donde la editaremos.

En primer lugar, vamos a configurar la forma de visualizar cada pista. Ahora podemos observar el nombre de pista y observaremos que cada una tiene un color distinto. Si queremos comprobar qué contiene cada una de una forma indicativa, elegimos la opción "Show events" en la opción "Part Appearance" del menú "Options".

Si deseamos tener más espacio para el visualizador de pistas, podemos ocultar la parte de información de pistas pulsando el botoncito aislado situado debajo de "Part Info", en la parte inferior izquierda de la ventana de arreglo.

Podemos observar que cada pista lleva asociada 8 casillas (ver fig. 5):

- **1. Casilla "A":** Indica la actividad de la pista.
- **2. Casilla "M":** Si la pista está mutada o no.
- **3. Casilla "C":** Indica el tipo de pista. Haciendo *clic* saldrá la lista con los distintos tipos.



Cada pista lleva asociada ocho casillas.

- **4. Casilla "Track":** Haciendo *clic* dos veces podemos asignar un nombre a la pista.
- **5. Casilla "Chn":** Indica el número de canal para la pista: 8 pistas mono para audio o 4 estéreo y 16 pistas MIDI.
- **6. Casilla "Output":** Aquí le indicamos al *Cubase* por qué dispositivo sonará la pista; la tarjeta de sonido (SB, ISIS) depende de la tarjeta o instrumento externo (Maestro, MPU o Módem, dependiendo de cómo tengas instalado el equipo y en qué interfaz esté conectado el instrumento)
- **7. Casilla "Instrument":** Sólo se activa en las pistas MIDI y sirve para anotar qué nombre de instrumento tiene asignado la pista. Se activa haciendo doble *clic*.
- **8. Casilla "T":** Si se activa, indicará que esa pista tiene los eventos fijados en el tiempo. Generalmente debe estar desactivada.

Con los cursores podemos desplazarnos por las pistas hacia arriba o hacia abajo. Si queremos que sólo suene la pista seleccionada pulsamos la tecla "*S*"; al desplazarnos, la pista activa seleccionada será la que suene sola y las demás se mutarán.



En el próximo número...

... aprenderemos los métodos básicos para manejar el *Cubase* y el uso de las herramientas de edición, así como a trabajar en la ventana de arreglo.

ZX Spectrum (I) De 1980 a 1984

¿Qué podemos decir de este ordenador?. Su creador, Sir Clive Sinclair, lanzó al mercado su primer Spectrum –el ZX-80– en 1980.

Estaba basado en el primer procesador de Zilog Z80, poseía la "enorme" cantidad de 1Kb de memoria y era monocolor. Fue destinado mayoritariamente a la enseñanza, ya que utilizaba el lenguaje *BASIC*. Sinclair Research Limited nació a raíz de comercializar el ZX-81, de iguales características que su predecesor, menor tamaño y teclas sensibles al tacto. El gran acierto de Sinclair fue situar sus precios por debajo de los de Atari o Commodore. Precisamente, la salida del fantástico Commodore VIC-20 obligó a Sinclair a lanzar el primer ordenador doméstico de ventas masivas en Europa, el ZX Spectrum de 16 K y 48 K, con un sistema de teclas de goma multifunción un poco particular y a veces odiosas. Aun así, la programación en código máquina de su microprocesador era relativamente fácil y cualquiera podría desarrollar juegos para esta máquina. Esta particularidad fue el trampolín que catapultó al ZX-Spectrum a la cima durante cerca de 10 años, ya que emergieron infinidad de casas desarrolladoras de juegos que producían una cantidad ingente de títulos diversos que las distribuidoras vendían a precios asequibles, inundando un mercado que devoraba y devoraba Spectrums.

La historia moderna que conocemos del mundo de los videojuegos la han forjado compañías y juegos que nacieron en la época de los 8 bits. Desde que hizo su aparición en Inglaterra el ZX 80 de Sir Clive Sinclair, los juegos para ordenador han adquirido otra dimensión en diversidad de géneros y tecnología. Sin embargo, no todo estaba inventado aún; en estos años los géneros que dominaban eran juegos de aventuras, arcades, matamarcianos, juegos deportivos o aventuras conversacionales. La tecnología de entonces no permitía un gran desarrollo tecnológico y el usuario estaba acostumbrado a estos géneros.

La llegada de los procesadores de 16 bits, principalmente al mundo de los PC de IBM, abrió la puerta a nuevas posibilidades técnicas, lo que contribuyó al desarrollo y evolución de nuevos géneros.

✍ AVANCES DE UNA DÉCADA

Sin más dilación, vamos a exponer los avances técnicos que sufrieron los juegos para ZX Spectrum a lo largo de su abultada historia. Mencionaremos sólo los títulos que han destacado por su tecnología.

✍ 1982

El Spectrum empezaba a conocerse en los hogares; todavía su programación y posibilidades eran desconocidas y comenzaban a aparecer juegos que versionaban las recreativas del momento como *Galaxian* o *Scramble* para promocionar la máquina. Quizás fue la adaptación del *Hobbit* el primer juego famoso para Spectrum. Nació por entonces la primera compañía de videojuegos importante, que se dio a conocer con el matamarcianos *Arkadia*. Nos estamos refiriendo a Imagine. En este año salió al mercado el primer simulador de vuelo para Spectrum,



1

Multitud de juegos de Spectrum se versioneaban para Amstrad.



2

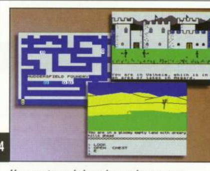
El ordenador de 8 bits más popular en Europa.

NOTA

¿Sabías que *Knight Lore* (1985) de Ultimate se programó antes que *SabreWulf* (1984)? Debido a su nueva tecnología y calidad, *Knight Lore* fue reservado para no ensombrecer el lanzamiento de *SabreWulf* (350.000 copias vendidas).



3
Dos fotos del ZX Spectrum, los ordenadores de 8 bits más populares de los años 80.


4
Unas capturas de los primeros juegos para Spectrum del año 1982.

DEFINICIÓN

▸ SPRITE EN EL SPECTRUM

Es un gráfico gestionado por una rutina especial que se puede mover por la pantalla sin alterar el fondo, dando la sensación de estar pasando por encima del mismo.

DEFINICIÓN

▸ TÉCNICA FILMATION

Esta técnica permitía realizar juegos en perspectiva isométrica en un Spectrum. Funcionaba dibujando las habitaciones en la pantalla en perspectiva isométrica, y por medio de sprites diseñados en diferentes ángulos daba la sensación de un entorno 3D. Después se desarrolló la técnica filmation 2, que consistía en mover los escenarios mediante scroll en vez de los personajes.

Flight Simulation de Psion; versión del primer *Flight Simulator* en blanco y negro del ordenador TRS-80 que posteriormente adquiriera Bill Gates para su serie de simuladores de vuelo.

1983

A partir de ese momento, prácticamente detrás de cada buen juego nacía una nueva compañía. Cabe destacar algunas como Incentive, Ocean o Bug Byte. Esta última desarrolló *Manic Miner*, quizás el primer juego de plataforma con niveles estudiados al milímetro, descubriendo un nuevo tipo de jugabilidad que pronto adquirirían multitud de desarrolladoras para sus juegos. También este año hace su aparición en escena la compañía más innovadora, sentando precedente en la creación de juegos para Spectrum: Ultimate Play The Game. Fueron sus miembros, sin duda, los creadores de los arcades de aventuras más fantásticos de la historia de los 8 bits. Títulos como *Atic-Atac* (en perspectiva cenital), *Sabre Wulf*, *Jet Pac* o *Pssst*, definieron el género, con gráficos detallados al máximo, montones de niveles y sin duda acción, mucha acción y en tan sólo 16 Kb de memoria.

1984

A partir de este año se inicia el *boom* del software para Spectrum. Las ventas decaían por la falta de juegos originales y las casas se tuvieron que poner las pilas. El género arcade shoot'em ups dominaba el mercado, pero el usuario pedía más calidad y nuevos géneros. La programación del Z80 ya no era un secreto y el mercado estaba inundado de libros y revistas especializadas. Esto contribuyó a explotar el rendimiento de los ZX Spectrum al máximo. Fue un año de muchos avances técnicos, y aún sorprenden los nuevos lanzamientos que parecían imposibles para el Spectrum. La calidad gráfica del Spectrum era tosca y el sonido se limitaba a un montón de pitidos monofónicos a veces molestos. Los colores en el Spectrum eran torpes de manejar, sólo se admitían 16 colores diferentes a la vez y 1 color por cada 8 píxels. Hasta la llegada de técnicas como la *Filmation* o la técnica para gestionar *sprites* no se pasaba de un mismo color para el fondo o de la misma perspectiva.

A finales de ese mismo año surge otra maravilla de Ultimate, *Knight Lore*. Primer juego que utilizó la técnica *Filmation* y que revolucionó el concepto de juegos para Spectrum. No tardaron mucho las demás desarrolladoras en engancharse al carro de la isometría.

Pero esto no termina aquí. Un escocés llamado Sandy White vuelve a sorprender a la comunidad del Spectrum con un juego que también utilizaba la perspectiva isométrica pero generada en 3D, *Ant Attack*. Después, desarrolló *I,of the mask* que utilizaba 3D real con objetos sólidos. Abundaban los simuladores espaciales en 3D con gráficos vectoriales como *3D Starstrike*, *Starion* o *Codename Mat*.

En este año también surgieron juegos que han hecho historia como *Avalon* de Hewson Consultant o la continuación del ya clásico *Manic Miner*, *Jet Set Willy* de Matthew Smith.

El género deportivo tuvo su mayor exponente en el que quizás sea el padre de todos los buenos juegos de tenis, sin lugar a dudas, el *Match Point* de Psion, que triunfó por tener unos gráficos simples pero cargados de realismo.

No queremos dejar este magnífico año sin hablar del juego que trajo el género *beat'em up* (peleas) a los ordenadores domésticos: nos estamos refiriendo al *Kung Fu* de Bug Byte.

En el próximo número...

... continuaremos este paseo por la historia de esta gran maravilla que fue el ZX Spectrum y que tan buenos ratos nos hizo pasar.

Cuestionario Videojuegos 4

Preguntas

1. Enumera las estructuras condicionales del Blitz3D para manipular el flujo de un programa.

2. Enumera las distintas estructuras para bucles del Blitz3D.

3. ¿Qué se entiende por protagonista principal de un juego?

4. ¿Qué aspectos de los personajes del juego tenemos que diseñar con detalle antes de modelarlos, animarlos y programarlos?

5. En el Milkshape3D enumera las teclas asociadas a las funciones de: seleccionar, mover, rotar y escalar.

6. Describe el procedimiento para reemplazar el color de una imagen en Paint Shop Pro.

7. En el programa GoldWave, ¿cómo podemos reproducir una selección tres veces seguidas?

8. ¿Cómo se selecciona un trozo del sonido en el GolWave?

9. ¿Qué significan las letras "M", "C" y las palabras "Chn" que aparecen en cada pista en la ventana de visualización de pistas del CUBASE VST?

10. Enumera las teclas asociadas a las siguientes funciones: reproducir, parar reproducción, ir a la posición del localizador izquierdo, ir a la posición del localizador derecho, ir al principio de la canción.

Respuestas al cuestionario 3

▷ **1.** Valores numéricos enteros, flotantes y valores alfanuméricos.

▷ **2.** **A.** Dim edificio (9,19)
B. Habitación = edificio (2,14)

▷ **3.** En un guión técnico se describen las facetas gráficas, sonoras y programación.

▷ **4.** Implementar un comportamiento imprevisible y con competitividad.

▷ **5.** Consiste en formar las caras del modelo utilizando triángulos, los cuales creamos previamente uniendo tres vértices.

▷ **6.** Es un visualizador previo de las imágenes contenidas en un directorio. Se activa con CTRL + B o eligiendo la opción "Browse" en el menú "File".

▷ **7.** Son aparatos que capturan sonidos externos y los transforman en audio digital para poder procesarlos.

▷ **8.** Pro Tools para Macintosh, Cubase para Atari y PC y AnvilStudio para PC.

▷ **9.** Hay que dibujar primero un cubo en el mapa y posteriormente elegiremos la geometría que queramos como un cilindro.

▷ **10.** Dibujamos un cubo que contendrá la luz. Una vez hecho esto pulsamos el botón derecho del ratón sobre la parrilla y elegimos la opción "Light".

Contenido CD-ROM 4

▶ AUDIO

■ Cubase VST
Nueva oportunidad para probar el magnífico programa de edición, con el cual podrás manipular tanto MIDI como pistas de audio.

■ Ashampoo AudioCD MP3 Studio 2000 Morpheus Edition 2.36
Programa para grabar cds y así salvar periódicamente tus composiciones para el juego.

■ Virtual Piano 3.0

Simulación de más de 600 instrumentos y otros interesantes efectos de sonido.

■ RioPort Audio Manager 3.5
Para buscar, grabar, y reproducir fácilmente toda clase de música digital.

■ Crakbone 1.1
Original herramienta con la que podrás convertir mágicamente palabras en melodías.

■ MySoundStudio 2.0
Aplicación muy completa con la cual crear, editar, y arreglar toda clase de música y efectos de sonido.

▶ DISEÑO 2D

■ Ulead Photo Explorer 7.0 Pro
Con este programa podrás organizar de un modo sistemático todas tus imágenes y contenidos multimedia.

■ FotoTime FotoAlbum 3.1.3

Otro excelente programa para organizar nuestras imágenes, cosa que deberemos hacer con eficiencia para controlar bien el diseño del juego.

■ Procreate Painter 7
Demo de la popular aplicación en su versión para Windows, con la cual podrás crear espectaculares ilustraciones y efectos.

■ Groone's KwikPik 2.11
Versátil herramienta para capturar pantallas de un modo rápido y eficiente.

■ XnView 1.21
Para ver, editar y convertir imágenes muy rápidamente podemos usar esta sencilla herramienta.

▶ DISEÑO 3D

■ Ulead Cool 3D 3.5
Con esta excelente herramienta podrás realizar títulos en 3D para luego utilizarlos en los créditos del juego.

■ Rhino 3D 2.0
Fantástico editor que permite crear cualquier tipo de objeto en tres dimensiones.

■ Amapi 3D Modeler 6.1
Otro potente editor interactivo para crear modelos 3D de un modo muy profesional.

■ Alice 99 2.01.005

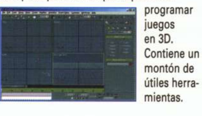

Desarrollo interactivo de mundos enteros en tres dimensiones.

■ DOGA-L1 9905
Especial para principiantes, esta aplicación nos enseña poco a poco a crear objetos en 3D.

▶ PROGRAMACIÓN

■ InterAx 1.2.0
Crea tus propios juegos de aventura en primera persona. Especial para no programadores.

■ VorteX 3.1
Programa juegos en RPG usando generadores de código, un depurador de código, y más...

■ Gmax 1.1
Completa plataforma que te permitirá

programar juegos en 3D. Contiene un montón de útiles herramientas.

■ Suneido 0.9.020313
Interesante programa que te permitirá mantener todas tus aplicaciones.

■ GOLD Parser Builder 1.0.20a
Potente herramienta que puede ser usada con numerosos lenguajes distintos de programación.

▶ JUEGOS Y EMULADORES

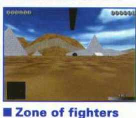
Emuladores:

■ Atari Classic Arcade
Nuevo emulador de Atari que te permitirá jugar a seis de los mejores juegos de tipo arcade en tu PC.

■ Amiga para PC
Especial para nostálgicos. Diviértete con este emulador que sin duda te enganchará.

Juegos:

■ Hyperspace Delivery Boy 1.0
Explora el universo y lucha contra monstruos y aliens en este entretenido juego de acción.

■ TankTime 3D 2.5

Excelentes gráficos 3D y sonidos en un juego que sin duda te cautivará.

■ Zone of fighters
Segunda oportunidad para obtener la versión mejorada de nuestro juego.

▶ VÍDEO

■ VScreen 1.0
Software para capturar vídeo y grabarlo en formato AVI.

■ Video Timecode Calculator 1.62
Potente complemento a los editores de vídeo, que te permitirá calcular el código de tiempo.

■ PowerDirector Pro 2.1

Captura y edita vídeo sin pérdida de calidad, con sencillez y rápidamente.